Secret Image Sharing Technique based on Bitwise XOR

Abstract—Traditional secret sharing schemes involve complex computation. A visual secret sharing (VSS) scheme decodes the secret without computation, but each shadow is much as big as the original. Probabilistic VSS solved the computation complexity and space complexity problems at once. In this paper we propose a probabilistic (2, n) scheme for binary images and a deterministic (n, n) scheme for grayscale images. Both use simple Boolean operations and both have no pixel expansion. The (2, n) scheme provides a better contrast and significantly smaller recognized areas than other methods. The (n, n) scheme gives an exact reconstruction. Due to rapid growth of internet, the secure transmission and protection of secret information has become an important issue. Numerous methods, such as cryptography and steganography have been developed to protect secure data from malicious users on the internet. But both the methods are Single Point of Failure (SPOF) type as they use single storage mechanism and thus these two methods are not robust against loss or manipulation. Secret sharing methods which distribute a secret content among a set of participants might be one of the possible solutions. This paper focuses on the major algorithms of the secret image sharing schemes.

INTRODUCTION

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image and the output of image processing may be either an image or, a set of characteristics or parameters related to the image. It may also be considered as a technique in which the data from an image are digitized and various mathematical operations are applied to the data, generally with a digital computer, in order to create an enhanced image that is more useful or pleasing to a human observer, or to perform some of the interpretation and recognition tasks usually performed by humans. Image processing usually refers to digital image processing, but optical and analog image processing are also possible. Analog Image Processing refers to the alteration of image through electrical means. The most common example is the television image. The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions digital image processing may be modeled in the form of Multidimensional Systems.

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements to the system. Only after the systems analysis we can begin to determine how and where a computer information system can benefit all the users of the system. This accumulation of the system is called a system study. In this phase we analyze the problem to get a clear understanding of the problem. we study the existing system and observe the problems present in it, and then try to recover from them. The system that we developing should overcome all the problems. The original image can be recovered only when any k of them are combined together, but any k − 1 or fewer shares cannot have sufficient information to reconstruct the original one. The original secret image, all n shares are required, any n − 1 or fewer cannot reconstruct a lossy or lossless version of the original secret image, i.e., Wang’s technique does not support the fault tolerance property which is the main requirement of secret sharing.

PROBLEM STATEMENT:

This paper addresses the area of secret sharing of image which is an application of image processing. There are several types of schemes developed all through the last two decades. Although there are several schemes available, none of them are efficient in obtaining the original image. But in many applications, the loss of information is deplorable and the image should be retrieved as delivered by source. In this paper we are introducing a new technique which even better in images thereby enhancing the efficiency in secret sharing by using Boolean operation.

PROPOSED SYSTEM:

In this paper, we implemented a secret sharing which overcomes the disadvantages of The aim of this paper is to improve the scheme proposed by W a n g et al. by developing a (k, n), 2 ≤ k ≤ n, secret image sharing scheme based on a Boolean operation with the same reconstruction complexity.

In this paper we have proposed a (k, n), 2 ≤ k ≤ n, secret image sharing scheme based on a Boolean operation with no reconstruction complexity.
ARCHITECTURAL DESIGN

Architectural Design involves identifying the software components, decoupling and decomposing them into processing modules and conceptual data structure and specifying relationships among the components.

Fig 1. Architectural Design of a system

Then final shadows for the color images are generated by composing the corresponding shadows from the Red, Green and Blue planes. Fig. 2 shows how to generate n share images from one color image.

Secret Sharing

A secret sharing scheme is a method via which a secret [2], commonly a cryptography key, is divided into multiple parts called shadows or shares and distributed to a collection of individuals or players. The agency responsible for performing the division is often called the dealer, and in many sharing schemes it is assumed that the dealer is perfectly honest. In the simplest schemes, the secret is divided into N shares and can only be recovered by gathering all N together at once; i.e. an (N, N) scheme. This has the advantage that it takes all of the participants to recover the secret, but if any of the shares are lost, the secret is also lost. For this reason, many secret sharing mechanisms are based on a (k, N) threshold mechanism in which the secret is divided into N shares, but can be totally recovered from any k (k < N) shares. Moreover, knowledge of k-1 shares should ideally provide no more information than that known from a single share (see definition of perfect sharing schemes). That is, the difficulty of any attack including brute force should be the same regardless of how many shares are known up until the threshold k is reached. These threshold mechanisms are weaker in terms of security than methods requiring all shares, but offer the ability to recover if several shares are lost. A final variant on secret sharing schemes occurs when you have a (k, N) threshold scheme but somehow divide the participants into authorized sets of k members such that the key can only be recovered if all the participants contributing shares actually form one of the authorized sets. Thus, not all sets of k shares must yield the key. It turns out that this last variant is the most general in scope as it encompasses the simple scheme mentioned at the beginning when k=N and the prior (k, N) scheme when all possible combinations of k shares form an authorized set; therefore, this scheme is actually used in many papers as the formal definition of a sharing scheme.

Functionalities

- Secret Image Sharing:
  - Sharing Phase
    - (n, n)-Threshold scheme
  - Selecting an image file
    - Selecting ‘n’, number of shares to generate
  - (r, n)-Threshold scheme
  - Selecting an image file
    - Selecting ‘r’ and ‘n’, numbers to generate shares
  - Recovery Phase
    - (n, n)-Threshold scheme
    - Selecting sufficient number of shares to reconstruct
    - secret image
    - (r, n)-Threshold scheme
    - Selecting sufficient number of shares to reconstruct
    - secret image
is represented in binary and the operation is carried out bit
by bit. For example, when \( a = 125 \) and \( b = 18 \), the XOR
between these two integers is
\[
a \oplus b = (125)_{10} \oplus (18)_{10} = (01111101)_{2} \oplus (00010010)_{2}
\]
\[
= (01101111)_{2} = (111)_{10}.
\]

For matrix inputs, the XOR operation of two \( NR \times NC \)
matrices is defined pixel-wise. That is,
\[
A \oplus B = [a_{ij} \oplus b_{ij}], \text{ where } i = 1, 2, \ldots, NR \quad j = 1, 2, \ldots, NC.
\]

The AND operation for integer scalar operands and matrix
operands can be defined similarly. Since I do not use the
Boolean OR operation and the Hamming weight, our
scheme is not a “visual” scheme and cannot be
implemented by directly viewing the stacked transparencies
of the shadow images. The pixel-wised Boolean operations
involved in our schemes can be easily carried out with
common software packages such as Photoshop. In all
calculations, every pixel is handled individually,
separated from other pixels. Therefore, when the context is
clear, I denote pixel \( A_{i} \) (s, t) simply as \( A_{i} \).

With the above construction procedure,
for a “0” pixel in \( A \) and any \( i \), I have \( C_{i} = B_{i} & 0 = 0 \) and
\( A_{i} = B_{n+1} \oplus C_{i} = B_{n+1} + 1 \), thus
\[A = A_{i} \oplus A_{j} = B_{n+1} \oplus B_{n+1} = 0.\]

For a “1” pixel in \( A \), \( C_{i} = B_{i} & 1 = B_{i} \) and \( A_{i} = B_{n+1} \oplus Bi \), thus
\[A = A_{i} \oplus A_{j} = B_{n+1} \oplus B_{n+1} \oplus Bi \oplus Bj = B_{i} \oplus Bj \]
which could be 0 or 1. In other words, between the original
image \( A \) and a reconstructed image \( A_{r} \), the “0” bits are kept
the same and the “1” bits may or may not change. With any
single shadow image, no information of \( A \) is revealed
because of the random nature of the matrices \( B \)’s. It is easy
to verify that the \( n \) matrices \( A_{1}, A_{2}, \ldots, A_{n} \) are \( n \) distinct
random matrices from construction method above;
each \( A_{i} \) \( (i = 1, \ldots, n) \) does not contain any information
of the original matrix \( A \).

**EXPERIMENTAL RESULTS**

This section presents the experimental results of the
proposed \((k, n)\) secret image sharing scheme. A \((2, 4)\) secret
sharing experiment is selected to demonstrate the
performance of the proposed method. A test image “Lena”
is used as a secret (input) image as shown in Fig shows the
generated noise like a shadow image using the proposed
method

### Peak signal-to-noise ratio

The phrase peak signal-to-noise ratio, often abbreviated
PSNR, is an engineering term for the ratio between
the maximum possible power of a signal and the power of
corrupting noise that affects the fidelity of its
representation. Because many signals have a very wide
dynamic range, PSNR is usually expressed in terms of the
logarithmic decibel scale. The PSNR is most commonly
used as a measure of quality of reconstruction of lossy
compression codecs (e.g., for image compression). The
signal in this case is the original data, and the noise is the
error introduced by compression.

When comparing compression codecs it is used as an
approximation to human perception of reconstruction
quality, therefore in some cases one reconstruction may

---

**THE \((2, n)\) SS SCHEME FOR BINARY IMAGES**

This section proposes a probabilistic \((2, n)\) SS scheme for
binary images. Boolean XOR and AND operations are
employed, and \(n+1\) distinct random matrices are generated
as intermediate results. Section 3.1 describes the shadow
image construction procedure. Section 3.2 compares this
scheme with Yang’s Prob VSS is described in a pseudo-
code style below in terms of its input, output, the
construction procedure (how to compute the shadow
images) and the revealing procedure (how to reconstruct
the secret image from the shadows).

**Input:** an integer \( n \) with \( n-2 \), and the secret image \( A \).

**Output:** \( n \) distinct matrices \( A_{1}, \ldots, A_{n} \), called shadow
images.

**Construction:** generate \( n+1 \) random matrices \( B_{1}, \ldots, B_{n+1} \); compute \( n \) intermediate matrices \( C_{1}, \ldots, C_{n} \) with
\( C_{i} = B_{i} \& A \) for \( i = 1, \ldots, n \); compute \( n \) shadow images \( A_{1}, \ldots, A_{n} \) with \( A_{i} = B_{n+1} \oplus C_{i} \) for \( i = 1, \ldots, n \).

**Revealing:** \( A = A_{i} \oplus A_{j} \), where \( i, j \in \{1, 2, \ldots, n\} \) and \( i \neq j \).

In the generation of the shadow images and in the
reconstruction of the secret, Boolean operations XOR (\(\oplus\))
and AND (\(\&\)) are used. For easy lookup, the truth-tables
of XOR and AND for binary scalar inputs are given below.

\[
\begin{array}{c|c|c}
\text{a} = 0 & \text{a} = 1 \\
\hline
\text{b} = 0 & 0 & 1 \\
\text{b} = 1 & 1 & 0 \\
\text{a} & \text{a} = \oplus \text{b} & \text{a} = \& \text{b} \\
\hline
\text{b} = 0 & 0 & 0 \\
\text{b} = 1 & 0 & 1 \\
\end{array}
\]

For integer scalar inputs between 0 and \( c - 1 \), each operand
is represented in binary and the operation is carried out bit
by bit. For example, when \( a = 125 \) and \( b = 18 \), the XOR
between these two integers is
\[
a \oplus b = (125)_{10} \oplus (18)_{10} = (01111101)_{2} \oplus (00010010)_{2}
\]
\[
= (01101111)_{2} = (111)_{10}.
\]
appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content. It is most easily defined via the mean squared error (MSE) which for two monochrome images $I$ and $K$ where one of the images is a noisy approximation of the other is defined as:

$$\text{PSNR}(\text{dB}) = 20 \log_{10} \frac{255}{\sqrt{\text{MSE}}}$$

Here, $\text{MAXI}$ is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with $B$ bits per sample, $\text{MAXI}$ is $2^B - 1$. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Alternately, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space, e.g., YCbCr or HSL. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. Acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25 dB. When the two images are identical, the same. For this value the PSNR is undefined (see Division by zero).

### Accuracy

The Peak Signal to Noise Ratio (PSNR) is applied to measure the quality of the reconstructed image. The higher PSNR indicates a better quality and lower PSNR denotes worse quality. The definition of PSNR is given in below.

The typical values for PSNR in a lossy image are within the range from 20 to 40 dB

$$\text{PSNR}(\text{dB}) = 20 \log_{10} \frac{255}{\sqrt{\text{MSE}}}$$

where MSE is the mean squared error between the original and the modified image which is defined as

$$\text{MSE} = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - I'(x,y))^2$$

Table 1. Comparison of PSNR of the reconstructed images of Chang et al. and the proposed scheme

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Gray scale image Max</th>
<th>Color image Max</th>
<th>Gray scale image Min</th>
<th>Color image Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>$\infty$</td>
<td>25.08</td>
<td>$\infty$</td>
<td>24.91</td>
</tr>
<tr>
<td>Chang et al.</td>
<td>33.07</td>
<td>33.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The PSNR of the reconstructed gray-scale and of the reconstructed color image is 33.70 and 33.75 respectively. On the other hand, the lowest and height PSNR of the proposed scheme are 24.91 and respectively. Table 1 shows the PSNR values of the proposed scheme and the scheme of Chang et al. [14-17]

### Analysis Of A Differential Attack

The Number of the Changing Pixel Rates (NPCR) and the Unified Average Changed Intensity (UACI) are designed to measure the resistance ability of the encrypted image against a differential attack. These two quantities are mathematically defined in following equations:

$$D(i, j) = \begin{cases} 0 & \text{if } C_1(i, j) = C_2(i, j) \\ 1 & \text{if } C_1(i, j) \neq C_2(i, j) \end{cases}$$

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{MN} \times 100\%$$

$$\text{UACI} = \frac{1}{M \times N} \sum_{i,j} \left| \frac{C_1(i,j) - C_2(i,j)}{255} \right| \times 100\%$$

where $C_1(i,j)$ and $C_2(i,j)$ are the gray-scale value of the original image and the encrypted image, respectively. The theoretical NPCR and UACI values of the image are 99.6094% and 33.4635%, respectively. The 99.6094% value of NPCR represents that the position of each pixel is dramatically randomized and the 33.4635% value of UACI indicates that the intensity levels of almost all pixels in the hared encrypted image are changed[17]. Table 2 shows that the average values of NPCR (> 99%) and UACI (33%) of the proposed method are very close to the theoretical values, which indicates that a tiny change in the original secret image will create a significant change in the encrypted (share) image.

Therefore, the encrypted shared images generated by our proposed scheme are robust against a differential attack.

Table 2. Values of NPCR and UACI tests of encrypted images of a gray image

<table>
<thead>
<tr>
<th>Test</th>
<th>Proposed</th>
<th>Chang (existed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPCR %</td>
<td>99.60</td>
<td>56.20</td>
</tr>
<tr>
<td>UACI %</td>
<td>32.23</td>
<td>56.20</td>
</tr>
</tbody>
</table>

Table 3. Values of NPCR and UACI tests of encrypted images of a color image

<table>
<thead>
<tr>
<th>Test</th>
<th>Proposed R</th>
<th>Proposed G</th>
<th>Proposed B</th>
<th>Chang (existed) R</th>
<th>Chang (existed) G</th>
<th>Chang (existed) B</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPCR %</td>
<td>99.45</td>
<td>99.65</td>
<td>99.65</td>
<td>70.10</td>
<td>70.10</td>
<td>70.10</td>
</tr>
<tr>
<td>UACI %</td>
<td>26.30</td>
<td>26.30</td>
<td>24.50</td>
<td>32.80</td>
<td>32.80</td>
<td>32.80</td>
</tr>
</tbody>
</table>

Table 3 shows the average values of NPCR (> 99%) and UACI (33%) for each component of the color images which are also close to the theoretical value. Hence, the proposed scheme for color images is also robust against a differential attack.

### Complexity Analysis

In[2] the computational complexity for the polynomial evaluation and interpolation is $O(k \log_2 k)$. Since Thien, and Lin have adopted Shamir’s $(k, n)$ scheme, their computational complexity for the recovery phase is the same as that of Shamir’s scheme, i.e. $O(k \log_2 k)$. Lin and Wang’s scheme is also based on the scheme proposed by Thien, and Lin, which raises the computational complexity to $O(k \log_2 k)$ for the recovery phase. The reconstruction process presented in this paper computes $n$ images using XOR and algebraic addition operations, resulting in
computational time proportional to $n$. The image construction is proportional to $k - 1$ because it includes XOR of all $n$ shares and addition of $k - 1$ shares. Therefore, the computational complexity is also dependent on the image size. So, computational complexity of $O(k)$, $k \leq n$ is established in this paper. Our method employs only arithmetic and Boolean operations rather than any geometric calculation, that is why it leads to low computational complexity. The methods related are compared with the proposed scheme in Table 4. The second row of Table 3 shows the comparison in terms of computational complexity of the proposed method and the related works.

Table 4. Comparison between the related image sharing and the proposed scheme

<table>
<thead>
<tr>
<th>Image scheme</th>
<th>Wang et al.</th>
<th>Our Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(k, n)$ secret Sharing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Reconstruction Complexity</td>
<td>$O(n)$</td>
<td>$O(k)$</td>
</tr>
<tr>
<td>Lossless secret Construction</td>
<td>Lossless</td>
<td>Lossy for $k&lt;n$ and Lossless for $k=n$</td>
</tr>
<tr>
<td>Fault tolerance Property</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

From Table 4 it is obvious that the reconstruction complexity of the method described in this paper is considerably lower than the one of the method described. Besides, the fault tolerance property of the method developed by us is better than that of the method[14]. Thus, considering both these properties, this research work is superior in the aspect that it includes both the properties which have not been included simultaneously in a single work.
CONCLUSION

A \((k, n)\) secret sharing scheme provides both binary image and natural image. In this paper we propose a new \((k, n)\) secret sharing scheme, based on a bitwise XOR. In the proposed scheme even if \(n - k\) shares are lost or corrupted, the remaining \(k\) shares are sufficient to recover the secret.

AND and XOR operations are used in the \((2, n)\) algorithm and only XOR operations are used in the \((n, n)\) algorithm. The proposed \((2, n)\) scheme is probabilistic and the contrast of the recovered image is 1/and \(2\), higher than other probabilistic \((2, n)\) schemes when \(n\) is greater than 2. Even, the reconstruction complexity of the method proposed is \(O(n)\) due to its bitwise XOR operation. Based on the Boolean operator XOR, the proposed scheme can easily recover the reconstructed image. Experimental results confirm that our proposed scheme not only gives high reconstructed image quality with a PSNR and MES These are the main advantages of our proposed scheme compared to the existing methods. Our secret sharing can also be applied on color images and it produces excellent results.

REFERENCES


Fig : reconstructed of image psnr % output